Galerkin/Linear Finite Elements Method in 1d, with non uniform coefficients

April 29, 2016

Problem Statement

We want to implement a solver for the following boundary value problem:

$$\begin{cases} -(a(x)u')' = f(x) & \text{in } (0,1) \\ u(0) = u(1) = 0 \end{cases} \tag{1}$$

where the coefficients $a(x)$ and $f(x)$ are to be specified by the user at run time

Weak Formulation

The weak form of equation (**??**) reads:

$$\begin{cases} \text{find } u \in V = H_0^1(0,1) s.t. \\ -\int\limits_0^1 (a(x)\, u')' \varphi \, dx = \int\limits_0^1 f(x) \cdot \varphi \, dx \quad \forall \varphi \in V \end{cases} \tag{2}$$

Using integration by parts we get

$$-\int\limits_0^1 (a(x)\, u')' \varphi \, dx = \int\limits_0^1 u' \varphi' \, dx - \varphi u' \Big|_0^1 = \int\limits_0^1 u' \varphi' \, dx$$

Galerkin Method

Let $V_h \subset V$ be a subspace of finite dimension N_h and let $\{\varphi_i\} \subseteq V_h$ be a basis of V_h

$$\begin{cases} \text{find } u_h \in V_h s.t. \\ \int\limits_0^1 a(x) u_h' \varphi_i' \, dx = \int\limits_0^1 f(x) \cdot \varphi_i \, dx \quad \forall \varphi_i \in V_h \end{cases} \tag{3}$$

We can express u_h as a linear combination of basis vectors:

$$u_h = \sum_{j=1}^{N_h} u_j \varphi_j \Rightarrow u_h' = \sum_{j=1}^{N_h} u_j \varphi_j'$$

Galerkin Method

Equation **??**$_2$ becomes:

$$\sum_{j=1}^{N_h} u_j \int_0^1 \varphi_i' \varphi_j' \, dx = \int_0^1 \varphi_i \, dx \quad i = 1, \ldots, N_h$$

We therefore get a linear algebraic system of the form

$$\mathbf{Au} = \mathbf{f},$$

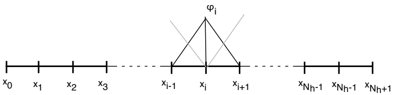
where

$$A_{ij} = \int_0^1 a(x) \, \varphi_i' \varphi_j' \, dx, \;\; f_i = \int_0^1 f(x) \, \varphi_i \, dx$$

and the vector of unknowns **u** is formed by the coefficients of the expansion of u_h w.r.t. the basis $\{\varphi_i\}$

Linear Finite Elements

Let us define a *triangulation* of the interval $(0, 1)$



Let as choose as the finite dimensional space V_h the set of functions that are continuous in $(0, 1)$ and are degree-1 polynomials (*i.e.*, affine functions) in each subintervall.
the canonical basis for V_h is gicen by:

$$\{\varphi_i\} = \{\varphi_i \in V_h \, t.c. \, \varphi_i(x_j) = \delta_{ij}\}$$

Implementation of the Linear FEM I

Using the local support property of the Finite Element basis $\varphi_i(x) \neq 0$ solo se $x \in (x_{i-1}, x_{i+1})$ we get

$$A_{ij} = \begin{cases} 0 \text{ if } |i-j| > 1 \\ \displaystyle\int_{x_{i-1}}^{x_i} a(x)\, \varphi_i'^2 + \int_{x_i}^{x_{i+1}} a(x)\, \varphi_i'^2 \text{ if } i = j \\ \displaystyle\int_{x_{i-1}}^{x_i} a(x)\, \varphi_i' \varphi_{i-1}' \text{ if } j = i-1 \\ \displaystyle\int_{x_i}^{x_{i+1}} a(x)\, \varphi_i' \varphi_{i+1}' \text{ if } j = i+1 \end{cases}$$

- the non zero terms in the matrix are $N_h + 2 * (N_h - 1)$ (the matrix is therefore sparse and tridiagonal)

Implementation of the Linear FEM I

Using the local support property of the Finite Element basis $\varphi_i(x) \neq 0$ solo se $x \in (x_{i-1}, x_{i+1})$ we get

$$A_{ij} = \begin{cases} 0 \text{ if } |i-j| > 1 \\ \int\limits_{x_{i-1}}^{x_i} a(x)\, \varphi_i'^2 + \int\limits_{x_i}^{x_{i+1}} a(x)\, \varphi_i'^2 \text{ if } i = j \\ \int\limits_{x_{i-1}}^{x_i} a(x)\, \varphi_i' \varphi_{i-1}' \text{ if } j = i-1 \\ \int\limits_{x_i}^{x_{i+1}} a(x)\, \varphi_i' \varphi_{i+1}' \text{ if } j = i+1 \end{cases}$$

- each entry in the matrix is given by a sum of (few) integrals each computed on only one subinterval

Implementation of the Linear FEM I

Using the local support property of the Finite Element basis $\varphi_i(x) \neq 0$ solo se $x \in (x_{i-1}, x_{i+1})$ we get

$$A_{ij} = \begin{cases} 0 \text{ if } |i-j| > 1 \\ \int\limits_{x_{i-1}}^{x_i} a(x)\, \varphi_i'^2 + \int\limits_{x_i}^{x_{i+1}} a(x)\, \varphi_i'^2 \text{ if } i = j \\ \int\limits_{x_{i-1}}^{x_i} a(x)\, \varphi_i' \varphi_{i-1}' \text{ if } j = i-1 \\ \int\limits_{x_i}^{x_{i+1}} a(x)\, \varphi_i' \varphi_{i+1}' \text{ if } j = i+1 \end{cases}$$

- the integrals cannot be computed exactly in general, we need to use a *quadrature rule*

Implementation of the Linear FEM II

To assemble **A**, we decompose the intervals of $(0,1)$ into integrals on subintervals:

$$A_{ij} = \int_0^1 a(x)\ \varphi_i'\varphi_j'\,dx = \sum_{k=1}^{N_h+1} \int_{x_{k-1}}^{x_k} a(x)\ \varphi_i'\varphi_j'\,dx$$

We can then use the following algorithm for assembling **A**:

1. Initialize all elements of **A** to 0
2. Compute integrals on subintervals
3. Compute entries od **A** as sums of the partial integrals

This is unnecessary for this simple case but has advantages in more complex situations we will discuss later:

1. Extension to different bases
2. Extension to multiple space dimensions
3. Parallel computing

Implementation of the Linear FEM III

In each subinterval (x_{k-1}, x_k) there are four integrals $\neq 0$ which need to be computed, they are usually arranged into a *local matrix*:

$$\mathbf{A_{loc}} = \left[\begin{array}{cc} i = k-1, j = k-1 & i = k-1, j = k \\ i = k, j = k-1 & i = k, j = k \end{array} \right]$$

$A_{loc_{11}}$ will then be added to $A_{k-1,k-1}$
$A_{loc_{12}}$ will then be added to $A_{k-1,k}$ etc...

this process is called assembly of the coefficient matrix

Exercise

- adapt the fem1d code to allow the user to specify the coefficients $a(x)$ and $f(x)$ at runtime
  - use the trapezoidal rule to compute integrals
  - use muparser to parse the function provided by the user
- adapt the fem1d code to allow the user to specify the quadrature rule at runtime